Centre de recherche
informatique de Montréal

A practical Petri net model for the state-oriented modelling of discrete event systems

Rapport technique
CRIM–92/08–05

Paul Freedman
Chercheur principal

Collection scientifique et technique
août 1992

Centre de recherche informatique de Montréal
3744, rue Jean-Brillant, bureau 500
Montréal (Québec)
H3T 1P1
Téléphone: (514) 340-5700
Télécopieur: (514) 340-5777

Tous droits réservés ©1992 CRIM
Dépôt légal – 3e trimestre 1992
Bibliothèque nationale du Québec
Bibliothèque nationale du Canada
ISBN 2-921316-06-4

ACKNOWLEDGEMENTS

This work was largely conducted while the author was a NATO Postdoctoral Science Fellow at the Laboratoire d'Automatique et d'Analyse des Systèmes (LAAS) du CNRS, Toulouse, France. The financial support of the French "Groupement Robots d'Intervention sur Site Planetaire" is gratefully acknowledged.

ABSTRACT

Many different Petri net (PN) models have appeared in the literature, but few are suitable for the 'state-oriented' modelling and analysis of a discrete event system. In this technical report, we present a detailed study of a particular place/transition PN we call Ω, and describe its relationship to the well-known PN classes of consistent nets, marked graphs, state machines, and free choice nets. We demonstrate that the property of strong connectivity is of critical importance for practical PN modelling of real world systems.

RÉSUMÉ

Le Réseau de Petri (RdP) est un outil graphique et mathématique qui s'applique à la modélisation et l'analyse de systèmes à événements discrets et asynchrones.

Dans ce rapport technique, on propose un nouveau modèle de RdP (place/transition) qui convient en particulier aux systèmes dont le comportement se définie sous forme de variables « d'états » qui changent leur valeurs. Par la suite, on fait le point sur ce nouveau modèle qu'on appelle Ω par rapport aux modèles de RdP bien connu dans la littérature tels que le graphe d'événements et la machine à état fini. De cette façon, on souligne que la modélisation de systèmes réalistes se présentent toujours comme les réseaux fortement connexes.

93-00269/

1. INTRODUCTION

Consider a discrete event system whose dynamics are described in terms of a single state variable[1] x and a set of possible values $X = \{x_1, x_2 \ldots\}$. The occurrence of an event gives rise to a change in state and x is assigned a new value $\in X$. Therefore, the behavior of the system may be represented by a trajectory in state space, i.e. $[x = x_1], [x = x_2] \ldots$

While many different Petri net models have appeared in the literature, few may be said to support such a *state-oriented* viewpoint.

One such model, the "superposed automata" (SA) net, was proposed in (De Cindio et al. 1979) for modelling non-deterministic sequential (computer) processes whose behavior is partially synchronized. Each process has private data and resources, and its behavior depends upon local conditions and its interactions with other processes. Places denote conditions of the form $[x = x_i]$, where x is a state variable and x_i a particular value. Syntactically, SA nets are a subclass of predicate/transition (Pr/T) nets with *balanced* transitions, i.e. for each transition, the numbers of input places and output places are equal.

A similar approach is presented in (Wilson and Krogh 1990). The authors cast a rule-based model of discrete event control into a PN which resembles the SA net, in that the pre-conditions and post-conditions of each rule (controller action) are composed of state variable conditions. However, the pre-conditions and post-conditions are not syntactically 'balanced' and as a result, this net model is not amenable to certain kinds of structural analysis. (Indeed, the paper concentrates on reachability analysis.)

A related model is presented in (Etessami and Hura 1991) which extends the rule-based model with timing and colouring. The resulting 'abstract PN' therefore makes it possible to parameterize each 'activity' (controller rule) in terms of arguments which may take on different values, as well as assign a (fixed) duration. Unfortunately, the paper concentrates on modelling, and the authors only hint at analytic methods.

We believe that 'practicality' means more than just modelling convenience and analytic power. For example, are markings always meaningful in the context of the problem ? Can an appropriate initial marking always be found ? Such questions are not addressed in any

[1] We shall not address the question of how to define an appropriate state variable formalism for a given problem. In the case of discrete event control, state variables typically have physical meaning and the ways in which their values change are governed by physical laws. In other contexts, the sense of state might be more abstract eg. a counter, the contents of a buffer, etc.

of three above-mentioned references.

In this technical report, we will address such issues in the context of a new Petri model we call Ω. We also describe its relationship to the PN classes of marked graphs, state machines, consistent nets, and free choice nets. Our study makes evident the importance of strong connectivity for PN models of real world systems.

2. PETRI NET PRELIMINARIES

Here we shall simply present key definitions and results required for our subsequent analysis. The reader unfamiliar with the general topic might wish to consult either (Peterson 1981) or (Murata 1989).

In the context of discrete event dynamic systems, it's well known that marked graphs are useful for modelling synchronization since the firing of a single transition may change the truth of multiple conditions. On the other hand, state machines are useful for modelling conflict associated with resource sharing. But the dynamics of real world systems are rarely modelled by either marked graphs or state machines. This suggests that a practical model must combine aspects of both. And since marked graphs and state machines are usually defined as sub-classes of place/transition (P/T) nets[2] we shall adopt this as our starting point.

In what follows, the following definitions shall be important (Murata 1989):

Definition 2.1 A P/T net is *strongly connected* if there exists a directed path (which respects the orientation of the arcs) in both directions (i) between any two places, and (ii) between any two transitions.

Definition 2.2 A *marked graph component* of a P/T net is a subnet generated by a subset of transitions $\{t_i\}$ such that (i) all of the input places and output places of each t_i is present, and (ii) each place in the subnet has at most one input transition and one output transition, i.e. the subnet induced by the transitions is a marked graph.

Definition 2.3 The integer solutions $\{c_i\}$ to the equation $Cc = 0$ are called *t-invariants*, where C is the $m \times n$ Incidence matrix of a net with m places and n transitions[3].

Definition 2.4 The transitions corresponding to the non-zero entries of a t-invariant are called its *support*, and those t-invariants with minimal support[4] shall be called *minimal* t-invariants.

[2] In the literature, the word 'system' is sometimes used to refer to a net and its marking. We prefer to use the older terminology, for which the word 'net' refers to the 'underlying net' plus its marking.

[3] In (Murata 1989), the Incidence matrix is defined to be the transpose of C as defined here.

[4] A support is minimal if no proper non-empty support of the support is also a support.

By combining these definitions, we may say that the strongly connected marked graph components of a strongly connected P/T net are the subnets generated by the minimal t-invariants.

It's easy to show that when a marked graph is strongly connected, then there is only one t-invariant $c = (11\cdots)'$. Conversely, when a state machine is strongly connected, each transition belongs to one of the strongly connected marked graph components generated by the minimal t-invariants.

Definition 2.5 A marked P/T net is *k-bounded* or simply *bounded* if the number of tokens in each place does not exceed a finite number k for any marking reachable the initial marking.

Definition 2.6 A marked P/T net is *safe* if it is 1-bounded, i.e. the number of tokens in each place is always 0 or 1.

Definition 2.7 A marked P/T net is *live* if no matter what marking has been reached from the initial marking, it is possible to ultimately fire any transition of the net by progressing through some further firing sequence.

The following results from (Murata 1989) relate strong connectivity and LS markings for P/T nets (Theorem 3, p. 555):

Theorem 2.1 If a connected P/T net has a LS marking, then it is strongly connected[5].

Theorem 2.2 For the case of marked graphs and state machines, strong connectivity is a necessary and sufficient condition for the existence of a LS initial marking.

We now turn to the class of P/T nets called *consistent* (Ramchandani 1974). As noted in (Ramamoorthy and Ho 1980),

> "Most real-world systems which function continuously with finite amounts of resources fall into the class of consistent systems."

[5]However, not every strongly connected P/T net has a LS marking.

Definition 2.8 A P/T net is *consistent* $\Leftrightarrow$ there exists an assignment of positive integer weights or 'currents' to all transitions such that for each place, the incoming and outgoing sums are equal (Ramamoorthy and Ho 1980).

Practically, this means that that the equation $Cc = 0$ has a solution $c \geq (11\cdots)'$. If the consistent net is also connected, then it is strongly connected, since there is just one t-invariant which contains all of the transitions.

3. Ω NET ANALYSIS

Let us now return to the state-oriented system modelling described in the Introduction. When the dynamics of a disrete event system are deterministic, the occurrence of an event gives rise to a predictable change in state, i.e. each event may be characterized by a state updating of the form $[x = x_i] \mapsto [x = x_j]$, where $x_i, x_j \in X$, the set of possible values. (Of course, there may be events whose occurrence does *not* change the value of a state variable, i.e. events of the form $[x = x_i] \mapsto [x = x_i]$.)

For our purposes, we shall consider events to be *actions* performed by a controllable agent and in this context, we shall call $[x = x_i]$ the action's *pre-condition* and $[x = x_j]$ the action's *post-condition*. When the system state is such that the pre-condition is true, we say that the action is enabled.

When the system dynamics are non-deterministic, an action may give rise to alternative changes in state, i.e. $[x = x_i] \mapsto [x = x_j]$ or $[x = x_i] \mapsto [x = x_k]$ where $x_i, x_j, x_k \in X$. Here we may say that the action has a single pre-condition but 'competing' post-conditions.

Of course, to describe the behavior of most systems, we typically require *many* state variables eg. $\{x, y\}$. Now an action may change the value of just one state variable or many state variables. For example, when the dynamics are deterministic, the action $[[x = x_i], [y = y_i]] \mapsto [[x = x_i], [y = y_j]]$, where the ',' denotes conjunction, signals a change in the state variable y, 'conditioned' by state variable x. In this sense, the occurrence of some events perform a partial (rather than complete) updating of the system state, by re-assigning new values to just some of the state variables. And as a consequence, we require that for each action, the state variables associated with the pre-conditions be exactly those state variables associated with the post-conditions. In previous work eg. (Freedman and Malowany 1988), we called this the "conservation of state variable assignments".

Remark: In order to simplify our exposition, we shall restrict the pre-conditions and post-conditions to be *conjunctions* of conditions. In this way, an action which may be enabled under different circumstances must be re-cast as a set of identical actions with slightly different labels, each enabled under a single set of circumstances. Since there is no disjunction, no state variable may appear more than once in the pre-conditions or

post-conditions of any action.

But whether the dynamics are deterministic or non-deterministic, two important assumptions will now be made[6]:

- **A1**: The system state is always uniquely defined, i.e. at any time, each state variable always has exactly one value[7].
- **A2**: System behavior is meant to be deadlock-free, i.e.the occurrence of deadlock represents a malfunction.

As a consequence, we may now present the following definition:

Definition 3.9 A state of a discrete event system is *meaningful* $\Leftrightarrow$ in this state and in all possible successor states, (i) each state variable has exactly one value, and (ii) there is no deadlock.

3.1 Preliminaries

As just described, the behavior of a discrete event system may be represented by a trajectory in state space, i.e. $[x = x_i], [x = x_j] \ldots$. Alternatively, behavior may be represented in terms of the changing truth values of the state variable, i.e. $x_i, x_j \ldots$. If we now associate a distinct place with each of these conditions and a distinct transition with each action, we may obtain a P/T net we call Ω with the following characteristics[8]:

- For each transition, the numbers of input places and output places are equal. We call such transitions *balanced*, after (De Cindio et al. 1979).
- For each transition, the state variables appearing in the conditions associated with the input places and output places are identical.

[6]For practical purposes, we shall also assume that no two actions have identical pre-conditions and post-conditions, since such actions would be indistinguishable.

[7]Such an assumption excludes from consideration stochastic (or possibilistic) dynamics for which a given state variable might have multiple values with varying probabilities (or possibilities).

[8]A precise definition of Ω will be presented shortly.

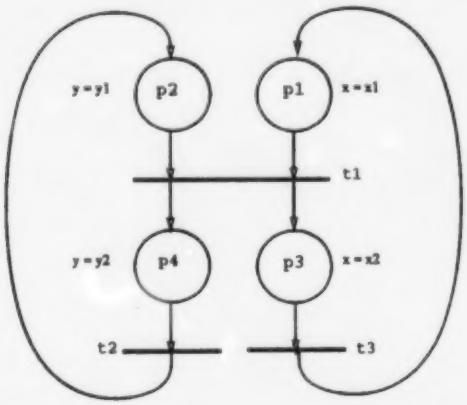

Figure 1: A simple Ω net which models the deterministic dynamics of two state variables.

One example of such an Ω net is shown in Figure 1. Each of the two state variables x, y may take on two different values, giving rise to a total of four conditions, hence four places in the Ω net. There are three actions, hence three transitions; the firing of t_2, t_3 give rise to the updating of just one state variable while the firing of t_1 gives rise to the updating of the complete system state, i.e. both state variables.

But how may we characterize those markings of the Ω net which correspond to *meaningful* states of the discrete event system ? The following theorem answers part of the question.

Theorem 3.3 Given an Ω net model of the dynamics of N state variables, then markings corresponding to meaningful system states must contain exactly N tokens such that each state variable has exactly one value.

Proof: By contradiction. If the marking assigns less than N tokens, then at least one state variable is not be assigned an value. Therefore, such an marking cannot represent a well formed state. If instead the marking assigns more than N tokens, then at least one state variable must be assigned multiple values. Therefore, such an marking cannot represent a well formed state. ■

To make the theorem easier to understand, return to the example shown in Figure 1. Consider first the marking $M_0 = (1000)'$ with one token. Clearly, M_0 cannot correspond to a meaningful system state because the state variable y is assigned no value, i.e. neither

place associated with y is marked. Now consider the marking $M_1 = (1110)'$ with three tokens. Here M_1 also cannot correspond to a meaningful system state because the state variable x is assigned two values, i.e. both places associated with x are marked.

In addition to this qualification on the number of tokens in a marking corresponding to a meaningful system state, we also require that markings be safe, since each place is meant to represent a condition. Finally, since deadlock may not occur, such markings must also be live. Indeed, we now demonstrate that it is the combination of safeness, liveness, and the number of tokens which together represent what is meant by meaningful, as per Definition 3.9.

3.2 Deterministic Ω nets

Not surprisingly, the class of Ω nets which model the dynamics of deterministic systems, forms a sub-class of marked graphs. This is because we require that our Ω nets be connected, and that all transitions be balanced.

Since meaningful states must correspond to net markings which are LS, it becomes important to discover under what circumstances marked graphs admit LS markings.

In the previous section, we noted that not every marked graph has a LS initial marking, since not every connected marked graph is strongly connected. Here we show that when the transitions are balanced, strong connectivity automatically follows.

Theorem 3.4 Every connected marked graph with balanced transitions is strongly connected.

Proof: By contradiction. Let G be an connected marked graph with balanced transitions which is *not* strongly connected. We may construct a connected digraph (directed graph) D from G by mapping transitions to nodes, and places to arcs.

Let us now follow the progress of a token circulating in D. A subnet S of D shall be called *absorbent* if it has no proper subnet such that once the token enters S, it cannot leave, i.e. it is forced to circulate within S and can never visit a node (transition) not in S. When considered as a 'black box', this means that there exists at least one arc from $(D - S)$ to S, but no arc in the opposite direction. Since the token *can* circulate freely in S, i.e.

the token may be guided to traverse any arc or visit any node, S itself must be strongly connected. (If this were not true, then S would simply contain an absorbent subnet, and could not be an absorbent subnet by itself.) Clearly, if G were strongly connected, then D would be strongly connected, and we would have $S \equiv D$.

But since D is not strongly connected, there must exist at least one such absorbent subnet S. This means that there exists at least one transition in G with multiple incoming arcs but just one outgoing arc, i.e. at least one transition which is not balanced. Hence, we have a contradiction. ■

The following corollary then follows from this theorem and Theorem 2.2.

Corollary 3.1 For every connected marked graph with balanced transitions, there exists a LS initial marking.

For the example shown in Figure 1, all four LS markings correspond to meaningful system states, and each marks exactly two tokens eg. $(1100)'$, corresponding to the system state $[x = x_1], [y = y_1]$.

But is it always the case that given a connected marked graph with balanced transitions, LS markings correspond to meaningful system states ? In the following example, we demonstrate that this is not always the case, since the number of tokens required for LS markings exceeds the number of state variables.

For the example shown in Figure 2, no assignment of a single value to the state variables x, y gives rise to a LS initial marking. Since the net is strongly connected, it must have a LS initial marking; indeed, two such markings are $M_1 = (10000110)', M_2 = (01000101)'$. But observe that each marking assigns exactly three tokens to the net; one to the state variable y, but two to the state variable x. Therefore neither LS initial marking corresponds to a meaningful initial state.

Why is this so ? Let us now take a closer look at the net structure. Consider, for example, an initial marking which marks places p_7, p_1, corresponding to $(x = x_2), (y = y_1)$. By firing transition t_1, we obtain $(y = y_2)$ and $(x = x_1)$. Then after firing t_5, we have $(x = x_4)$ and at this point, no transition is enabled; in order to fire t_2, we require that $(x = x_3)$ but in order to make $(x = x_3)$, we must fire t_3 which in turn requires that $(y = y_3)$. Indeed, the reader may verify that deadlock occurs for all possible assignments of one initial value to x, y and this is why LS initial markings must mark three tokens instead of two.

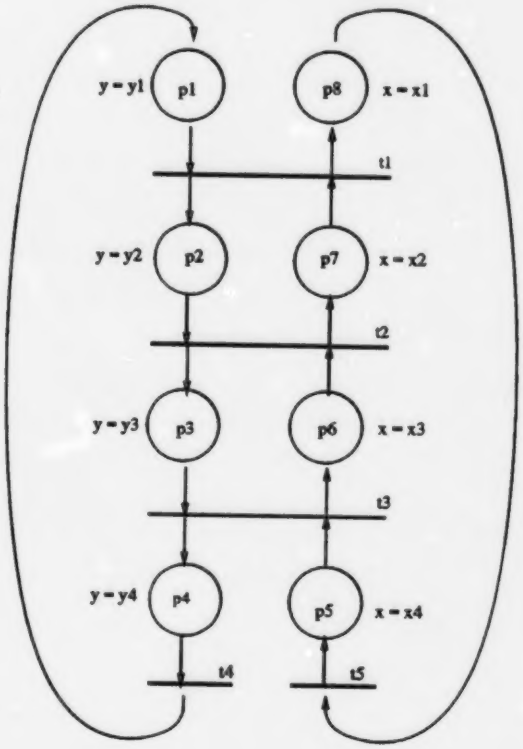

Figure 2: A connected marked graph with balanced transitions which is not an Ω net.

More generally, the fact that there exists no LS initial marking corresponding to a meaningful initial state, indicates that something is wrong with the state variable description of the discrete event system (given the assumptions presented in Section 2). For this example, the inter-linking of the various actions (transitions) gives rise to an incoherence in the state variable semantics.

We are now in a position to precisely define the class of Ω nets which model the dynamics of deterministic systems.

Definition 3.10 An Ω net model of the deterministic dynamics of N state variables, is a connected marked graph with balanced transitions for which LS markings contain exactly N tokens such that each state variable has exactly one value.

3.3 Non-deterministic Ω nets

Now we shall move beyond the case of deterministic behavior to define a certain kind of restricted conflict for which our deterministic results may be extended. More precisely, we shall define a class of P/T nets with non-deterministic firing behavior which may be decomposed into the kind of deterministic Ω nets described in the previous section. In this way, the behavior of the non-deterministic Ω net may be studied in terms of the behavior of its deterministic 'components'.

Consider the non-determinism associated with the outcome of an action eg. success or failure. This may be modelled by associating a distinct 'version' of the action with each of the possible outcomes. Since the different versions share the same set of pre-conditions, they are, by definition, mutually exclusive. But we shall also require that the consequences of these versions be eventually 'unified' by a another set of conditions, so that the original 'conflict' may re-occur. In previous work, we called such a logical structure *choice among alternatives* (Freedman and Malowany 1988).

To make these ideas clearer, consider the case of the connected state machine. Observe that the transitions of connected state machines are, by definition, balanced, since each place has exactly one input transition and exactly one output transition. In the example shown in Figure 3, transitions t_1, t_3 may be said to 'compete' for the updating the value of state variable x. Observe that place p_4 serves to unify the consequences of firing t_1 or t_3, so that this 'competition' may re-occur.

More precisely, the competing (deterministic) behaviors of the discrete event system may be studied via the (connected) marked graph components associated with the various ways in which the conflict may be resolved. For this example, there are two minimal t-invariants: $c_1 = (11001)', c_2 = (00111)'$ which generate the marked graph components shown in Figure 4.

Definition 3.11 Given a connected P/T net with balanced transitions in conflict. The (connected) marked graph components induced by the minimal t-invariants shall be called the *decomposition* of the conflict.

Definition 3.12 Given a connected P/T net with balanced transitions in conflict. Such conflict represents an instance of ***choice among alternatives*** $\Leftrightarrow$ the decomposition into

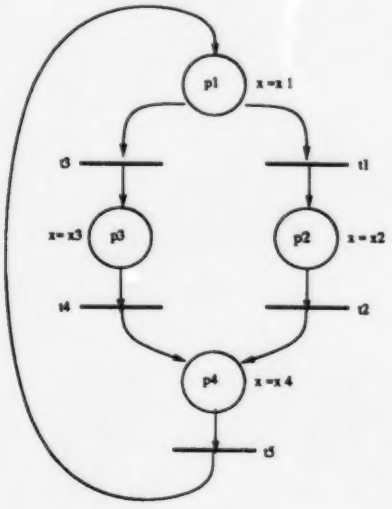

Figure 3: A simple Ω net which models the non-deterministic dynamics of one state variable.

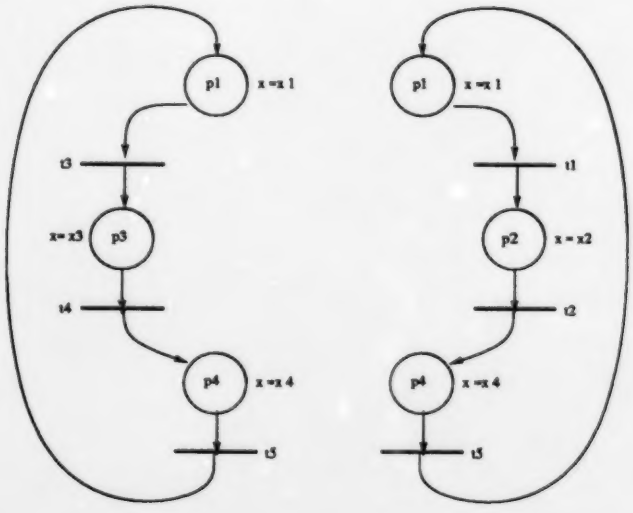

Figure 4: The two (connected) marked graph components associated with the competing behaviors of the Ω net in the previous figure.

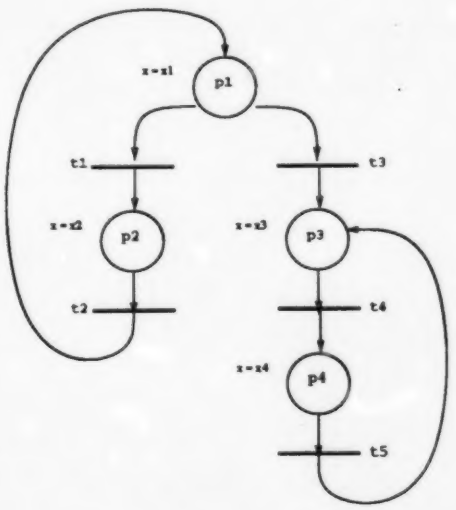

Figure 5: A connected state machine which is not an Ω net.

(connected) marked graph components is such that (i) every transition belongs to some component, (ii) no two transitions in conflict appear in the same component.

In words, when the non-determinism in the original net is an instance of choice as defined here, the decomposition into the connected marked graph components effectively identifies the alternative behaviors associated with the conflict. If the conflict cannot be decomposed in this way, then it does not represent an instance of choice, and one such example is shown in Figure 5. Here the consequences of the two 'competing' transitions t_1, t_3 are not unified. As before, there are two minimal t-invariants $c_1 = (11000)', c_2 = (00011)'$ but transition t_3 is not covered. Therefore, the marked graph components cannot truly represent the alternative behaviors associated with the conflict. As we shall see, what is 'wrong' with this state machine is the fact that it is connected, but not strongly connected.

We may now extend the previous definition of deterministic Ω nets to the class of Ω nets associated with the the restricted kind of non-determinism we call choice among alternatives.

Definition 3.13 An Ω net model of the non-deterministic dynamics of N state variables,

is a strongly connected P/T net with balanced transitions and instances of choice among alternatives, for which LS markings contain exactly N tokens.

The following theorem follows from the definition.

Theorem 3.5 Every Ω net is strongly connected.

Proof: Since all Ω nets are connected P/T nets, and since we are always able to determine a LS initial marking, then all Ω nets are strongly connected (Theorem 2.1). ■

We are now in a position to state and prove the following result.

Theorem 3.6 Every strongly connected state machine is an Ω net.

Proof: Let us look more carefully at the decomposition into connected marked graph components. Since the state machine in connected, each marked graph component is also strongly connected. Since the state machine is strongly connected, it is covered by these components. Therefore, the non-determinism of the state machine may be viewed in terms of instances of choice among alternatives. ■

One example of a non-deterministic Ω net which is neither a marked graph nor a state machine, is shown in Figure 6 with its decomposition in Figure 7.

3.4 Free choice nets and Ω

Free choice nets are a well-known class of P/T nets which also combine properties of marked graphs and state machines. Free choice nets require that if two transitions share a common input place, then there are no markings for which one transition is enabled and not the other. In this way, when the common input place is marked, there is a 'free' choice about which transition to fire.

It's well-known that a LS free choice net is covered by strongly connected marked graph components (Murata 1989), (Theorem 14, p. 557). This suggests a similar strategy for decomposing the non-determinism associated with the system dynamics, as we have presented for Ω nets.

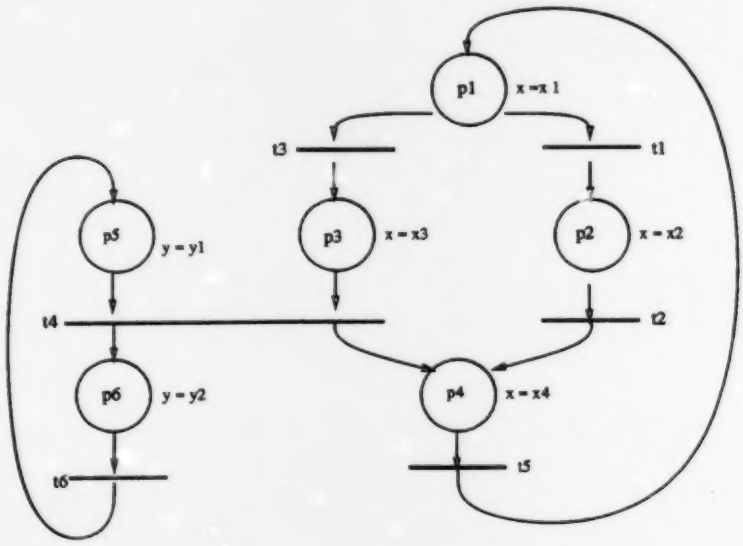

Figure 6: An Ω net which combines a single instance of choice with synchronization.

It's easy to see that free choice nets and Ω nets have some overlap; there are nets which belong to both classes, and nets which belong to just one. For example, free choice nets whose transitions are not all balanced are not Ω nets. Conversely, the Ω net shown in Figure 8 is not a free choice net. Here the input place p_2 associated with the condition $[b = b_1]$ is shared by transitions t_1, t_2, but the presence of the other input places p_1, p_3 associated with the conditions $[a = a_1], [c = c_1]$, means that these transitions are not enabled under identical circumstances.

3.5 Consistent nets and Ω

Unlike the comparison with free choice nets, Ω nets may be seen as a *sub-class* of consistent nets, since connected consistent nets form the class of strongly connected P/T nets,

Clearly, any consistent net for which the arc weights are not all 1, is is not an Ω net[9].

There are other differences. Whereas places in Ω nets may only contain one token (since

[9]Indeed, the presence of arc weights > 1 'removes' in some sense the restriction that transitions be balanced.

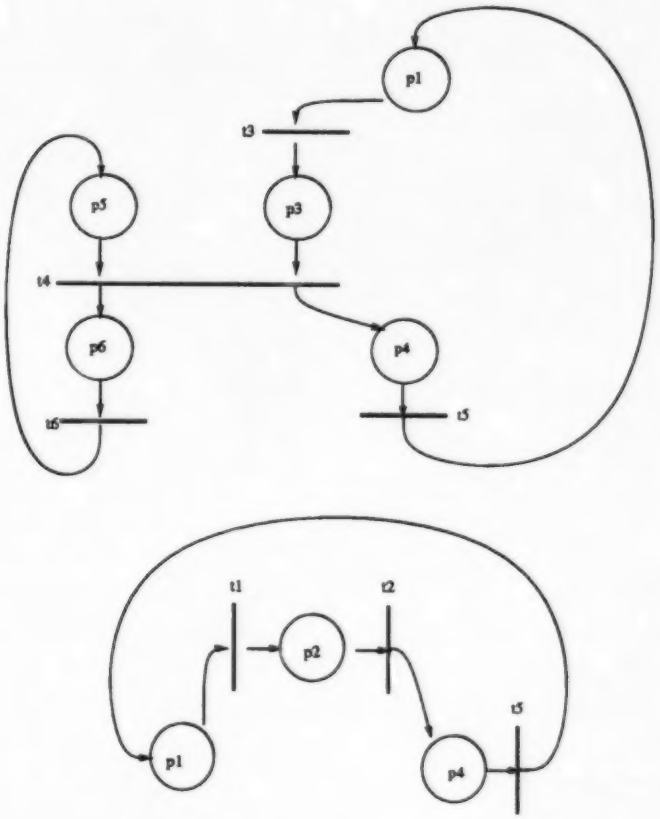

Figure 7: The decomposition of the non-deterministic dynamics of the Ω net shown in the previous figure.

they are associated with true/false conditions), places in consistent nets may contain multiple tokens. Therefore, for the class of consistent nets, we require that markings be bounded; for the class Ω, we require that markings be safe. For this reason, consistent nets which require live markings to be bounded rather than safe, do not belong to the class Ω eg. Figure 2.

We therefore conclude this comparison with the schematic shown in Figure 9.

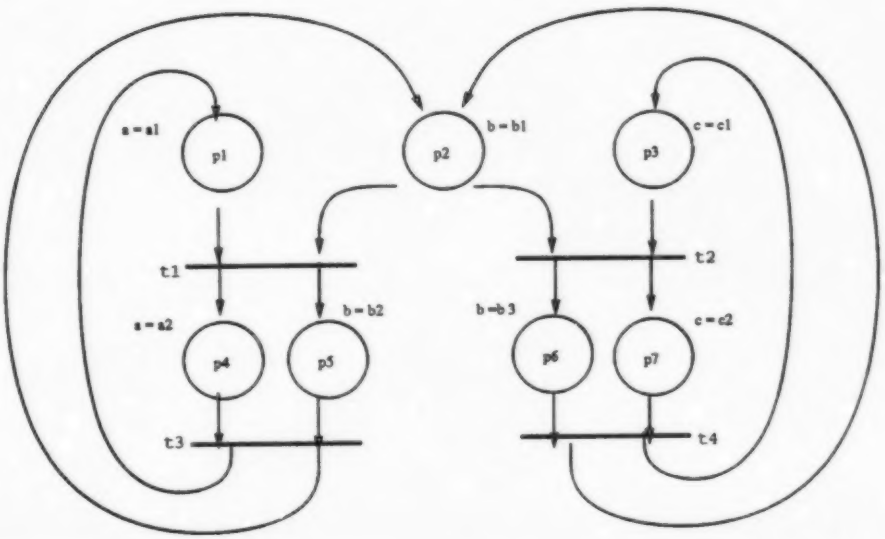

Figure 8: An Ω net which is not a free choice net.

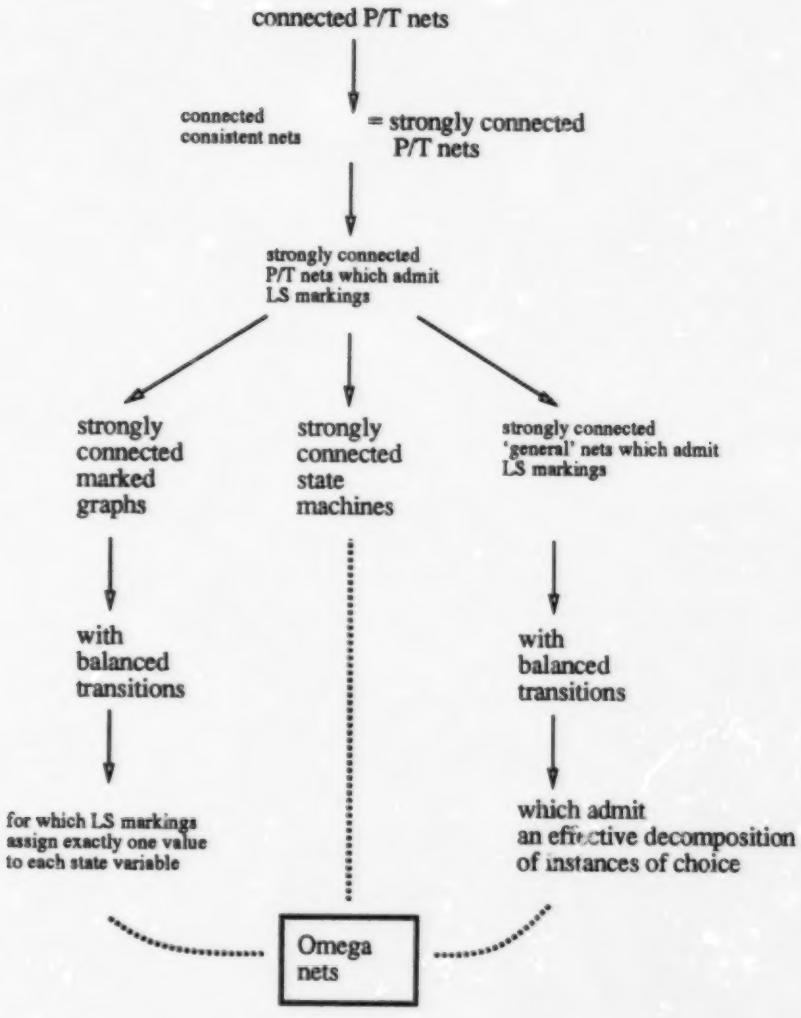

Figure 9: Comparing various classes of strongly connected P/T nets.

4. CONCLUSIONS

The dynamics of many problems in discrete event control may be described in terms of state variables and how their values change when actions (or more generally, events) take place. Although many different Petri net models have appeared in the literature, few are suitable for such state-oriented modelling and analysis.

In this technical report, we defined a particular kind of strongly connected P/T net called Ω for which actions are characterized by *pre-conditions* and *post-conditions*, each defined by conjunctions of conditions of the form $[x = x_i]$ where x is a state variable and x_i a particular value. As a consequence, all transitions are balanced, i.e. each one has equal numbers of input places and output places.

We showed that Ω nets associated with dynamics which are deterministic, form a sub-class of strongly connected marked graphs. In addition, we defined a restricted kind of non-determinism called *choice among alternatives* which may be studied in a decomposed form in terms of marked graph components. Indeed, such decomposition provides an effective way of determining whether or not a given connected P/T model of a state-oriented description of the system dynamics, is an Ω net.

Like free choice nets, the class Ω may therefore be seen as combining the determinism modelled by (strongly connected) marked graphs with the non-determinism modelled by (strongly connected) state machines. But unlike free choice nets, which we demonstrate to be unsuitable for the state-oriented modelling we deal with here, Ω nets may only model a restricted kind of non-determinism for which the consequences of each instance of conflict are 'unified', so that the conflict may re-occur.

Ω nets do form a sub-class of connected consistent nets, known to be important for modelling systems which function in continuous way with finite amounts of resources. Whereas a given consistent net may or may not admit a live and bounded (LB) marking, Ω nets by definition always admit live and safe (LS) markings. Elsewhere (Freedman 1991), we present algorithms to identify those states which correspond to LS markings suitable for initializing the operation of the system.

REFERENCES

De Cindio, F., G. D. Michelis, L. Pomello, and C. Simone. 1979. Superposed automata nets. In *Second European Workshop on Application and Theory of Petri Nets (Informatik Fachberichte no. 52)*. Springer Verlag.

Etessami, F., and G. Hura. 1991. Rule-Based Design Methodology for Solving Control Problems. *IEEE Trans. on Software Engineering* 17(3):274–282.

Freedman, P. 1991. Ω: a new Petri Net model for the state-oriented modelling and analysis of discrete event systems. Technical Report 91123, LAAS/CNRS, April.

Freedman, P., and A. Malowany. 1988. The analysis and optimization of repetition within robot workcell sequencing problems. In *IEEE Int. Conf. on Robotics and Automation*, 1276–1281, Philadelphia.

Murata, T. 1989. Petri nets: Properties, analysis, and applications. *Proceedings of the IEEE* 77(4):541–580.

Peterson, J. 1981. *Petri Net Theory and the modeling of Systems*. Prentice-Hall.

Ramamoorthy, C., and G. Ho. 1980. Performance evaluation of asynchronous concurrent systems using petri nets. *IEEE Trans. on Software Engineering* 6(5):440–449.

Ramchandani, C. 1974. *Analysis of Asynchronous Concurrent Systems by Petri Nets*. PhD thesis, L.C.S./M.I.T., February.

Wilson, R., and B. Krogh. 1990. Petri net tools for the specification and analysis of discrete controllers. *IEEE Trans. on Software Engineering* 16(1):39–50.